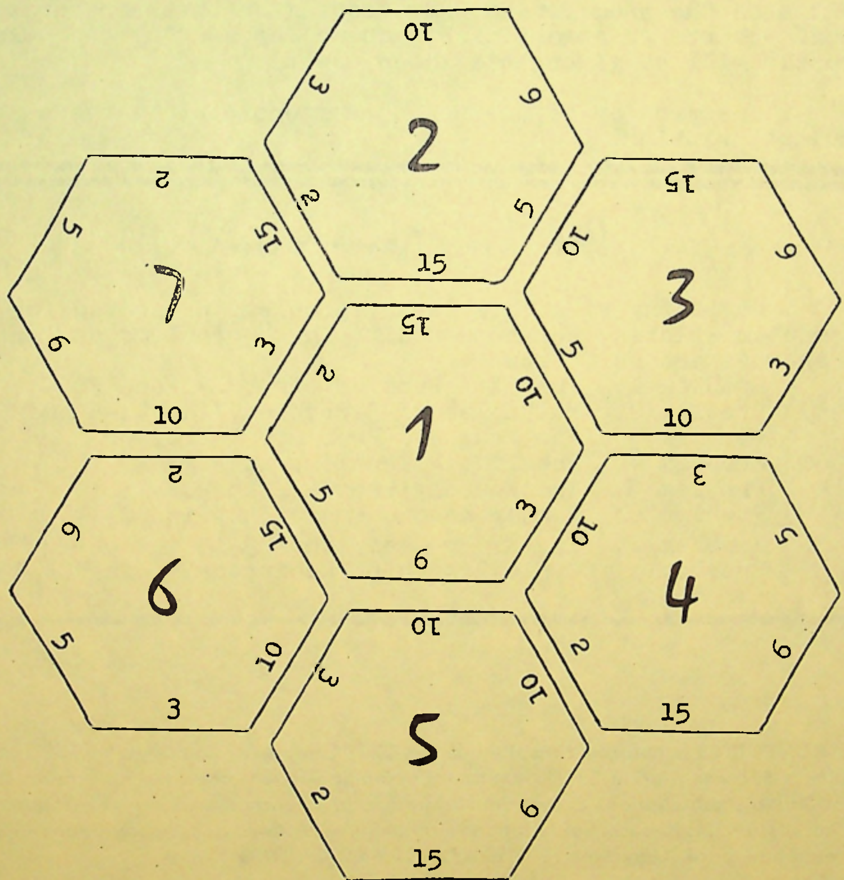
Popular Computing VOL 1 NO 5 AUG 1973

BOX 272
CALABASAS,
CA. 91302



The Hexagon Problem

PROBLEM 9

The Hexagon Problem

The seven hexagons shown in the Figure are to be rearranged in position and rotation so that the pairs of numbers at the adjacent edges have no factor in common except 1. In the pattern as shown, this is true for the edges between hexagons 2 and 7, and 5 and 6, but is not true for the edges between hexagons 2 and 3, or 4 and 5.

There are $6! \cdot 6^6$ possible arrangements of the hexagons (i.e., 33,592,320 arrangements) but not all of them need to be tested. For any given arrangement, there are 12 pairs of numbers to test for a greatest common divisor of 1, but again, not all 12 tests need to be made, since any failure vitiates the need for further testing on that arrangement. A flowchart for an efficient scheme for solving the hexagon puzzle will be given in a later issue.

The problem is due to Lee Morgenstern; it has a unique solution.

Problem Numbering

Beginning with this issue, problems presented for computer solution are numbered. The 8 problems so far presented are as follows:

- Problem 1. The 3X+1 Problem, in PC1-1 and PC4-6.
- Problem 2. Multiples of 3 Problem, in PC1-5 and PC4-8.
- Problem 3. The Four 4's Problem, in PC2-3.
- Problem 4. The Zigzag Problem, in PC2-13.
- Problem 5. The Fourway Problem, in PC3-1.
- Problem 6. Numbering the Fractions, in PC3-9.
- Problem 7. The A^B Problem, in PC3-13.
- Problem 8. Cycle Length of Reciprocals, in PC4-13.

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $15 per year, or $12 if remittance accompanies the order. For all foreign subscriptions, add $4.50 per year. Multiple subscriptions to the same address, add $5 each; thus, 3 copies per month is $25 per year, U.S. delivery. Back issues $1.50 each. Subscriptions may begin with any issue. Subscriptions for qualified undergraduate students half price.

5

Log 5	0.6989700043360188047862611052755069732318101185379
Ln 5	1.6094379124341003746007593332261876395256013542685
$\sqrt{5}$	2.2360679774997896964091736687312762354406183596115
$\sqrt[3]{5}$	1.7099759466766969893531088725438601098680551105431
$\sqrt[5]{5}$	1.3797296614612148323900634642160176928556498779776
$\sqrt[7]{5}$	1.2584989506418267349927871711777138948186804815067
$\sqrt[10]{5}$	1.1746189430880190059144636656918989016737230187574
$\sqrt[100]{5}$	1.0162245912673256358168906791588194469598007476627
e^5	148.41315910257660342111558004055227962348766759388
π^5	306.01968478528145326274131004343560648030070662807
$\tan^{-1} 5$	1.3734007669450158608612719264449611486509995958997
5^{100}	7888609052210118054117285652827862296732064351090230047702789306640625
5^{1000}	933263618503218878990089544723817169617091446371708 024621714339795966910975775634454440327097881102359 594989930324242624215487521354032394841520817203930 756234410666138325150273995075985901831511100490796 265113118240512514795933790805178271125415103810698 378854426481119469814228660959222017662910442798456 169448887147466528006328368452647429261829862165202 793195289493607117850663668741065439805530718136320 599844826041954101213229629869502194514609904214608 668361244792952034826864617657926916047420065936389 041737895822118365078045556628444273925387517127854 796781556346403714877681766899855392063601631001004 211973674701749862626690747296762535803929376233833 981046927874558605253696441650390625

N-SERIES

Book Review

A GUIDE TO FORTRAN IV PROGRAMMING (Second Edition) by Daniel D. McCracken, John Wiley & Sons, 1972, soft cover, 288 pages, $6.95.

Reviewed by Edward A. Ryan, Woodland Hills, California

The catalog of the library of books that have earned the accolade of "classic" in their field would be a small one, indeed. McCracken's A GUIDE TO FORTRAN IV PROGRAMMING would most certainly be a member of that library. In the short span of time since its introduction, McCracken's book and its FORTRAN II predecessor have outsold all other texts on the subject of FORTRAN combined. More importantly, the book appears to have withstood the test of time, as is clearly evident in the recently published second edition.

Anyone familiar with the first edition will immediately recognize the clear, crisp, readable style of writing that distinguishes this book from the stodgy, language manual style of so many of its competitors. The new edition is nearly double the size of the previous version. The organization and presentation of the material follows that of the earlier version quite closely, however (once you have found a formula for success, why change it?).

Much of the added volume comes from additional examples and case studies, as well as additional exercises. The text accompanying the presentation of examples is so clear and unambiguous that in many cases I was able to anticipate the coded FORTRAN statements prior to looking at them. A great deal of the expanded volume comes from new material dealing with the interactions of FORTRAN in the time sharing environment, although I was disappointed not to see an example of a FORTRAN application designed specifically for this medium.

McCracken and his associates are to be congratulated for the job they did in assembling and editing the manuscript for the second edition.

The new book is printed on high quality, easy-on-the-eyes, non-glare paper and utilizes easily readable type. Throughout the book I was able to find only two small, insignificant typographical errors and these were in areas where you wouldn't ordinarily be looking for such errors. The flowcharts used within the book, for the most part, follow the ANSI standard. The only major criticism I can make of the book is that many of the figures and tables, sample programs, etc., appear on pages other than those containing the text which refers to them.

McCracken's new book has much to offer both the neophyte just entering the computing industry and the experienced professional who has been around for some time. Of greatest importance to beginners is the fact that the book will impress him with the fact that FORTRAN is a flexible, general purpose problem solving tool and is not strictly the toy of the scientist or engineer.

Book Review

COMPUTING WITH MINI COMPUTERS
by Fred Gruenberger and David Babcock
Melville Publishing Company, Los Angeles, xv + 288 pps.
$11.95 hardcover

Reviewed by R. W. Hamming

The mini computers have arrived. Most of the officialdom of computing has long been wrapped up in the giant machines, but a few alert people have noticed the minis. Fortunately Gruenberger and Babcock have done something about the lack of textbooks in computing that recognize the mini as a worth while machine on which to learn about computing. Their text does not distract the beginner from the essence of computing by getting a huge mass of hardware plus a vast, ill-documented mess of software systems between the user and computing.

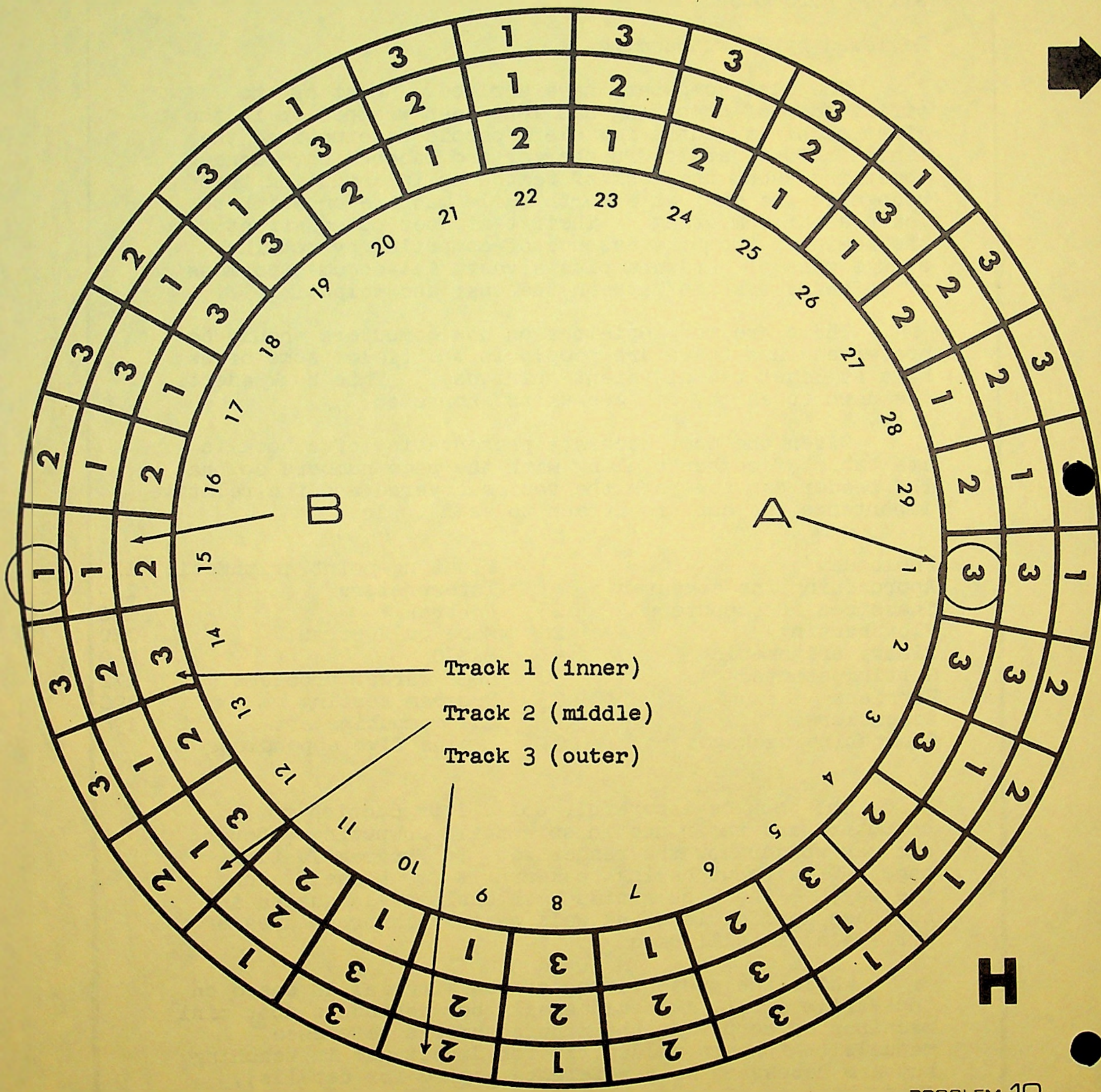
There are more opinions on how computers should be presented than there are people in the field (some of us have at least two different opinions). This book adopts the down to earth, no "gee-whizz" approach.

Often the most accurate presentation of a book is its table of contents along with the page numbers so that the reader can see both the topics covered and the relative importance the authors attach to each topic.

The book is a careful, solid introduction to computers and their use in scientific computing (though one may wonder how the reader gets to Program Testing, page 190, without having to learn it for himself along the way). The book avoids both business languages and business applications, as well as the glamorous area of artificial intelligence.

While the presentation of the material is based on the Varian 620/L, it could easily be adapted to most mini machines by supplying the manufacturer's reference manuals (which are seldom fit for direct use in teaching but are necessary as a reference source for details). Thus, because minis are becoming so widespread, this book fills a real need.

Three Track



H

Two markers, A and B, move in the circular pattern shown in Figure H. Marker A is about to move three sectors forward, to sector 4. Marker B is about to move one sector forward, to sector 16. These two moves initiate the procedure.

Subsequent moves are made as follows. Marker A is at sector 4, track 1; marker B is at sector 16, track 3. A will move on the track specified by B (in this case, track 2), which takes A to sector 6, track 2. B then moves on track 3 (as specified by A) from sector 16 to sector 18. For each move, B's position dictates the track to be used by A, and A's position then dictates the track to be used by B.

What, then, is the logic of a subroutine which, when called, outputs a full move; that is, the next position for both A and B? Given that logic, and a debugged and tested program for it, it should be possible to predict the positions of the markers at any subsequent time. To provide a specific goal (and to be able to compare independent results), we will seek the positions of the two markers (sector and track) at the end of 1000 full moves.

The use of the subroutine will make it possible to ascertain whether or not A ever overtakes or passes B, or vice versa.

Desk Calculator Review

Hewlett-Packard HP-80

The HP-80 is the "commercial" version of the HP-35. In place of the keys that are dear to scientists and engineers (logarithms and trigonometric functions), the significant functions on the 80 are for compound interest, statistical calculations, and a calendar. The machine uses floating decimal arithmetic and will expand into scientific notation, although there is no mechanism for entering numbers in scientific notation directly. The number of decimal places desired in the display (up to 8) is programmable. The number of functions available on the machine is increased over the HP-35 by use of a shift key; that is, several of the keys have two purposes.

The chief utility of the machine lies in its logic for compound interest. Of the four factors entering such a calculation (the number of time periods; the interest rate per time period; the payment per period; and the future value), any three may be entered and the machine will furnish the fourth. Thus:

$17 at 6%/year for 23 years yields $64.94.
$17 per year at 6%/year for 23 years yields $798.93.
$17 at 6%/year yielding $100, the interest rate is 8.008%
$17 per year for 100 years at 100% yields $2.1155E31

Some compound interest calculations can cause the machine to cycle for long periods of time. For example, $17 per year for 100 years to yield $10000: the interest rate is 2.95% and the calculation takes 15 seconds. If the future value in that calculation is changed to $1,000,000, the interest rate is 1431.28% and the calculation takes 30 seconds. By stepping up the future value and the number of time periods, the machine can be made to cycle indefinitely, and can only be stopped by shutting off its power.

For those calculations of compound interest that were within range, results were checked against Financial Compound Interest and Annuity Tables, Financial Publishing Company, 1960, 884 pages.

The calendar buttons will calculate elapsed days between two dates within a 200 year range. Thus, entering January 1, 1901 and December 31, 1970, the result is 25566 days. Or, entering January 1, 1901 and 25566, the result is December 31, 1970. All the test cases listed on pages 29 and 33 of Computing: A Second Course (Gruenberger, 1969) were in agreement with the results on the HP-80.

Means and standard deviations are readily calculated; the standard deviation formula uses N(N-1) in its denominator (i.e., sample deviation). The Trend Line key gives a linear regression for data at equally spaced intervals.

The HP-80 sells for $395. For anyone who deals in money calculations, the machine is a marvel.

Desk Calculator Review

Canon F-10

The Canon F-10 ($545) is a 12-digit desk machine with a large clear display, weighing 6 pounds and measuring 10 1/4 x 11 1/2 x 3 1/2, AC only. It has floating arithmetic (but not scientific notation) plus 0, 1, 2, 3, 4, 5, and 6 positions of fixed point. Function keys include square and cube root, squaring, common and natural logarithms, e^x and a^x, sine, cosine, tangent, inverse trigonometric functions, degrees to radians and radians to degrees, rectangular to polar coordinates and the inverse, and degrees, minutes, and seconds to decimal degrees.

All the functions are calculated by iteration. At least 10-digit accuracy is claimed for all functions; this seems to be correct. The normal tests:

ln x followed by e^x to produce x
sin x followed by arcsin x to produce x
cube root x followed by cube x to produce x
$\sin^2 x + \cos^2 x$ to produce 1.

all checked to at least 10 significant digits for a wide range of x values. Other interesting results were:

$$\ln 2.71828182845 = 1.$$
$$\log 10 = 1.$$
$$\log 100 = 2.$$
$$\log 200000000 = 8.301029996$$
$$\tan^{-1} 1 = 45.$$
$$\sin 60 = .8660254035$$

97, sqrt, sqrt, sqrt, sqrt, sqrt, sq, sq, sq, sq, sq = 96.99999681

rectangular $\sqrt{3}$, 1 to polar = ρ = 2., θ = 30.00000001

The trigonometric function keys operate only in degrees; sin 36030 = .5, and so on.

The function $\sqrt{A}^{\sqrt{B}}$ was explored at length; see results elsewhere in this issue.

Of the desk-top machines so far reviewed (the Victor 18-1721 in PC-4 and the Dietzgen ESR-1 in this issue), the Canon F-10 is outstanding. It is well designed, reasonably priced, and performs exactly as advertised.

Desk Calculator Review

Dietzgen ESR-1

The advertising brochure for the Dietzgen ESR-1 ($695) states "every operation, from simple multiplication to the most complicated chain calculation, is completed in seconds, to 12 digit accuracy." The speed claim is excessive; for example, the machine has a cube root key; the operation is performed by some interval-halving scheme and takes 15 seconds for one cube root.

Natural logarithm and e^x are available; for the latter function, the brochure states "Exponent--Automatically computes exponent of number displayed." On entering the number 3, and then taking the natural logarithm followed by the exponential, the result is 2.99999998898, which hardly conforms to the claim of 12 digit "accuracy." Common logarithms are available, but not the inverse function of 10^x. The natural logarithm of 2.71828182845 comes out .996314418; the common logarithm of 10 as .999999998.

The brochure states "Arc Functions--Automatically computes arc functions of any number." The only arc function on the machine is arctangent.

Again, the key a^n is described as "Nth power--Raise to any power, automatically." The action of the key is severely limited; by "any power" is meant integral values of n, from 2 to 9.

Keys are available for hyperbolic sine and cosine; for degrees to radians; and for radians to degrees. Conversion of degrees, minutes, and seconds to decimal degrees is a built-in function. The machine also has two storage registers.

The Dietzgen machine can be switch-programmed for floating decimal operation (but not scientific notation) or 2, 3, 4, 6, or 8 positions of fixed decimal. The claim is made for "Automatic Round-off--rounds off the last digit displayed if five or greater," whatever that means.

The machine is badly designed, inaccurate in its calculations, priced outrageously, and touted in fatuous terms. A Mark 2 machine is said to be pending, for which the Dietzgen company should do its homework.

A Note on the A^B Problem

In PC3-14 there was given a table of certain values for the expression A^B, where the values of A and B were the square roots of small primes. Some of the entries given in that table are reproduced in the accompanying chart, recalculated several ways. The low order 6 significant digits of the table from PC3 are compared as calculated on four desk calculators and in double precision Fortran.

		2	3	5	7	11	13
	P	526919	634654	509877	642535	470775	908194
	Q	526918	634654	509877	642535	470774	908194
2	R	526918	634654	509877	642535	470778	908197
	S	526919	634654	509877	642535	470775	908195
	T	526919	634655	509879	642537	470778	908200
	P	055549	412396	650954	832911	031865	036964
	Q	055549	412395	650954	832911	031864	036964
13	R	055549	412395	650954	832911	031878	036965
	S	055545	412377	650953	832909	031855	036962
	T	055549	412386	650954	832910	031861	036963
	P	612109	029437	296751	431002	415774	879364
	Q	612109	029437	296750	431002	415774	879363
29	R	612109	029437	296750	431002	415782	879368
	S	612110	029434	296753	431008	415774	879363
	T	612110	029435	296755	431006	415774	879365
	P	840502	848093	358797	768390	648657	918535
	Q	840502	848093	358797	768389	648656	918535
31	R	840502	848093	358797	768389	648665	918544
	S	840503	848090	358800	768399	648658	918538
	T	840503	848092	358804	768401	648660	918542
	P	014455	916446	062249	357105	474137	968683
	Q	014454	916446	062248	357105	474137	968682
43	R	014454	916446	062248	357105	474157	968700
	S	014454	916441	062248	357104	474137	968678

P: Results on an early HP-35 not corrected for the defect mentioned on page 6 of PC-1

Q: Results on an HP-80

R: Results on a later model HP-35

S: Results on a Canon F-10, reviewed in this issue

T: Results in double precision Fortran (to at least 13S)

Correction

On page 13 of PC-3 there appeared a discussion of irrational numbers that lie close to integral values. Mr. Herman P. Robinson, of the Lawrence Berkeley Laboratory, has pointed out several errors on that page.

The value of A^B, where A and B are irrationals, is not necessarily irrational. The simplest counter example is $A = e$, $B = \ln 2$.

Further, Mr. Robinson notes that values of irrationals can be made arbitrarily close to integral values with ease. For example, $\cos(10^{-6})$ differs from unity by 10^{-18}.

More interesting is the expression $(1 + \sqrt{2})^N$, which gets closer to an integral value as N increases. The table below summarizes some calculated results.

A further erratum: the value of $\sqrt{19}^{\sqrt{11}}$ was given on page 14 of PC-3 as 131.9997009, but was copied incorrectly on page 13.

The problem of finding values of A and B such that $\sqrt{A}^{\sqrt{B}}$ lies close to an integer is still open.

N	Number of digits to the left of the decimal point in $(1 + \sqrt{2})^N$	Number of 9's after the decimal point.
1	1	0
2	1	0
4	2	1
8	4	3
16	7	6
32	13	12
64	25	24
128	49	48
256	98	97
512	196	195
1024	392	391

The last value is reproduced on the next page, as calculated with 1000-digit precision.

The 100-square Trip Problem

PROBLEM 11

In the course of a larger problem, a subroutine is needed to traverse the pattern shown. The input to the subroutine is a triplet (R,C,N) indicating the Row and Column of a cell, and the Number of cells to advance; the output is a pair (R,C) indicating the cell at which the trip terminates. For any trip that ends outside the pattern, the subroutine should furnish the pair (11,10). Some test cases are:

(10, 1, 100)	(11,10)
(10, 1, 99)	(10,10)
(8, 4, 17)	(5, 6)
(1, 1, 1)	(1, 2)
(1, 1, 20)	(1, 3)
(10, 2, 1)	(10, 3)
(5, 6, 50)	(11,10)
(10, 2, 19)	(9, 4)
(10, 2, 39)	(9, 6)
(6, 3, 57)	(9, 9)

$(1+\sqrt{2})^{1024} =$

916857381894481845645133291190504333843461823250529940377136
344149075342541907081048361452797154017773822687232789944875
430563534335657529154975938905684991897404007678416106442939
995558475666390154545677008456626198312155556295338963185799
267110365897313197677934242730037120092270362251009811020913
614836914671675304114971866015244505554124859414396428293601
20406934959087847500441905528833.999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999
999989093178287622853...

The logic of the subroutine could be satisfied by having a table in storage:

10	1	
9	1	1
8	1	2
7	1	3
6	1	4
5	1	5
4	1	6
3	1	7
2	1	8
1	1	9
1	2	10
2	2	11
3	2	12
.	.	..
.	.	..
1	6	50
2	6	51
3	6	52
.	.	..
.	.	..

4	9	86
3	9	87
2	9	88
1	9	89
1	10	90
2	10	91
3	10	92
4	10	93
5	10	94
6	10	95
7	10	96
8	10	97
9	10	98
10	10	99
11	10	100

and this information could be contained in 101 computer words (if the word size is 22 bits or greater), although it would probably be more expedient to have it in 303 words, for ease in manipulating it.

It would seem that the problem could be attacked more logically than by table lookup. A flowchart for this subroutine is needed.